# Scope of Work: Sole Power Mobile App Development

**Table of Contents**

### 1. Overview

This Scope of Work (SOW) outlines the development of a custom mobile application for the Town of Vail's Sole Power program. The app will serve as the primary platform for participants to track their human-powered commutes, engage with the program, and contribute to the overall goal of reducing vehicle miles traveled and greenhouse gas emissions within the Eagle County community. The app will be designed to be user-friendly, engaging, and accessible to a wide range of users, including those who may not have access to a compatible mobile device.

### 2. Executive Summary

App Maisters will develop a robust and intuitive mobile application that seamlessly integrates with the existing Sole Power program. Our approach will focus on creating a user-centric experience that encourages active participation and promotes a sense of community among participants. We will leverage modern technologies and best practices in mobile app development to ensure a high-quality, secure, and scalable solution. The app will be built with a focus on:

- **User-friendly interface:** The app will feature a clean and intuitive design that is easy to navigate and understand, regardless of the user's technical expertise.
- **Comprehensive functionality:** The app will include all the features outlined in the RFP, including trip tracking, goal setting, leaderboards, challenges, rewards, and social sharing.
- **Data security and privacy:** We will implement robust security measures to protect user data and ensure compliance with all relevant regulations.
- **Scalability and adaptability:** The app will be designed to accommodate future program expansions and updates, such as the potential integration of bus ridership and carpooling tracking.

### 3. Functional Hierarchy of Roles

This section outlines the functional hierarchy of roles within the Sole Power mobile app, providing a structured overview of the app's functionalities and how they are organized based on user roles.

### 3.1. Participant Role

- Account Management
  - Sign Up
  - Log In
  - Log Out
  - Profile Management
  - Team Management
- Trip Tracking & Logging
  - Trip Log Portal
  - Trip Tracking
  - Manual Trip Entry
- Data & Analytics
  - Personal Dashboard
  - Leaderboards
  - Season Dashboard
  - Lifetime Dashboard
- Navigation & Information

    - Mapping & Geolocation
    - Program Information
    - Contact Information
- Social Features
    - Social Sharing
    - Peer-to-Peer Encouragement
    - Team Chat
- Push Notifications
    - Program Updates
    - Challenge Reminders
    - Personal Progress Updates
- Accessibility
    - Language Support
    - User Interface Design

### 3.2. Administrator Role

- User Management
    - User List
    - User Search
    - User Edit
    - User Deactivation
    - User Reactivation
- Team Management
    - Team List
    - Team Creation
    - Team Approval
    - Team Edit
    - Team Deletion
- Trip Management
    - Trip List
    - Trip Search
    - Trip Edit
    - Trip Deletion
- Data & Analytics
    - Dashboard Access
    - Data Export
    - Data Analysis
- System Management
    - Push Notification Management
    - Content Management
    - System Settings
- Security & Access Control
    - Role-Based Access Control
    - User Authentication
    - Data Encryption
- Integration
    - Website Integration
    - Third-Party Integrations

## 4. Functional Details of Roles

This section provides detailed descriptions of the functionalities associated with each user role, outlining the specific actions and interactions that users can perform within the app.

### 4.1. Participant Role

#### 4.1.1. Account Management

- **4.1.1.1. Sign Up:**
    - **Summary:** This task involves collecting essential user information for account creation, including first name, last name, email address, password, and password confirmation. Input validation is implemented to ensure data integrity and security.
    - **Description:** The sign-up form will collect essential user information, including first name, last name, email address, password, and password confirmation. All fields will be validated to ensure data integrity and format compliance. For instance, the email address will be checked for a valid format and uniqueness within the system. The password will be subject to complexity requirements, such as minimum length and inclusion of specific characters. Secure input methods will be implemented to prevent Cross-Site Scripting (XSS)

attacks.

- **4.1.1.2. Account Creation:**
  - **Summary:** This task involves generating a unique user ID, hashing the password, storing user data securely in a database, and implementing access control mechanisms.
  - **Description:** Upon successful validation, a unique user ID (UUID) will be generated for the new user. The user's password will be hashed using a secure hashing algorithm like bcrypt, ensuring that the original password is not stored in plain text. User data will be stored securely in a database, with encryption at rest implemented to protect the data from unauthorized access. Access control mechanisms will be implemented to prevent unauthorized access to user data.
- **4.1.1.3. Welcome Message:**
  - **Summary:** This task involves displaying a welcome message to the user after successful account creation, providing a brief overview of the app's features and guiding the user to the next step.
  - **Description:** After successful account creation, a welcome message will be displayed, providing a brief overview of the app's features and functionalities. The message will guide the user to the next step, such as logging in or exploring the app.
- **4.1.1.4. Log In:**
  - **Summary:** This task involves collecting the user's username/email and password, implementing input validation, and using secure input methods to prevent XSS attacks.
  - **Description:** The login form will collect the user's username/email and password. Input validation will be implemented to ensure data integrity and format compliance. Secure input methods will be used to prevent XSS attacks.
- **4.1.1.5. Authentication:**
  - **Summary:** This task involves verifying the entered username/email against the database, hashing the entered password, comparing it to the stored hashed password, and implementing rate limiting to prevent brute-force attacks.
  - **Description:** The entered username/email will be verified against the database. The entered password will be hashed and compared to the stored hashed password. Rate limiting will be implemented to prevent brute-force attacks.
- **4.1.1.6. Session Management:**
  - **Summary:** This task involves generating a unique session ID, storing it securely, and implementing session timeout and inactivity logout mechanisms.
  - **Description:** Upon successful authentication, a unique session ID (JWT) will be generated. This session ID will be stored securely in a cookie or local storage. Session timeout and inactivity logout mechanisms will be implemented to enhance security.
- **4.1.1.7. Access Control:**
  - **Summary:** This task involves redirecting the user to the appropriate dashboard based on their role and implementing authorization mechanisms to restrict access to specific functionalities.
  - **Description:** The user will be redirected to the appropriate dashboard based on their role (Participant). Authorization mechanisms will be implemented to restrict access to specific functionalities based on user roles.
- **4.1.1.8. Log Out:**
  - **Summary:** This task involves deleting the session ID from the cookie or local storage and invalidating the user's session on the server-side.
  - **Description:** The session ID will be deleted from the cookie or local storage. The user's session will be invalidated on the server-side.
- **4.1.1.9. Redirect:**
  - **Summary:** This task involves redirecting the user to the login screen or the app's home page.
  - **Description:** The user will be redirected to the login screen or the app's home page.
- **4.1.1.10. Profile Management:**
  - **Summary:** This task involves displaying the user's profile information and implementing access control to ensure that users can only view their own profile information.
  - **Description:** The user's profile will display their information, including first name, last name, email address, team (if applicable), date of birth (if provided), and phone number (if provided). Access control will be implemented to ensure that users can only view their own profile information.
- **4.1.1.11. Edit Profile:**
  - **Summary:** This task involves allowing users to update their profile information, implementing input validation, and using secure input methods to prevent XSS attacks.
  - **Description:** Users will be able to update their profile information, including first name, last name, email address (subject to validation), password (subject to complexity requirements), date of birth, and phone number. Input validation will be implemented to ensure data integrity and format compliance. Secure input methods will be used to prevent XSS attacks.
- **4.1.1.12. Password Reset:**
  - **Summary:** This task involves allowing users to request a password reset via email, sending a password reset link, and implementing a secure password reset process.
  - **Description:** Users will be able to request a password reset via email. A password reset link will be sent to the user's email address. A secure password reset process will be implemented, ensuring that the new password meets complexity requirements.
- **4.1.1.13. Email Address Change:**
  - **Summary:** This task involves allowing users to change their associated email address and implementing a secure email verification process.
  - **Description:** Users will be able to change their associated email address. A verification email will be sent to the new email address. A secure email verification process will be implemented.
- **4.1.1.14. Team Management:**
  - **Summary:** This task involves displaying a list of available teams, implementing pagination and search functionality, and allowing users

to filter teams.
  - **Description:** A list of available teams will be displayed, including team name, team description, and number of members. Pagination and search functionality will be implemented for large team lists.
- **4.1.1.15. Join Team:**
  - **Summary:** This task involves allowing users to join a team, implementing a confirmation step, and updating the user's profile with the new team affiliation.
  - **Description:** Users will be able to join a team by selecting it from the list. A confirmation step will be implemented to ensure the user's intent. The user's profile will be updated with the new team affiliation.
- **4.1.1.16. Create Team:**
  - **Summary:** This task involves allowing users to create a new team, implementing input validation, and submitting the team creation request to the administrator for approval.
  - **Description:** Users will be able to create a new team by providing a unique team name and a team description. Input validation will be implemented to ensure data integrity and format compliance. The team creation request will be submitted to the administrator for approval.
- **4.1.1.17. Leave Team:**
  - **Summary:** This task involves allowing users to leave their current team, implementing a confirmation step, and updating the user's profile with the new team affiliation.
  - **Description:** Users will be able to leave their current team. A confirmation step will be implemented to ensure the user's intent. The user's profile will be updated with the new team affiliation (or team affiliation will be removed if leaving all teams).

#### 4.1.2. Trip Tracking & Logging

- **4.1.2.1. Trip Log Portal:**
  - **Summary:** This task involves displaying a list of all logged trips, implementing pagination and search functionality, and allowing users to filter trips.
  - **Description:** A list of all logged trips will be displayed, including date, miles, trip type, mode, start time, end time, start location (latitude/longitude), and end location (latitude/longitude). Pagination and search functionality will be implemented for large trip lists. Users will be able to filter trips by date, trip type, or mode.
- **4.1.2.2. Edit Trip:**
  - **Summary:** This task involves allowing users to modify existing trip details, implementing input validation, and using secure input methods to prevent XSS attacks.
  - **Description:** Users will be able to modify existing trip details, including date, miles, trip type, mode, start time, end time, start location, and end location. Input validation will be implemented to ensure data integrity and format compliance. Secure input methods will be used to prevent XSS attacks.
- **4.1.2.3. Delete Trip:**
  - **Summary:** This task involves allowing users to remove a trip from their log and implementing a confirmation step.
  - **Description:** Users will be able to remove a trip from their log. A confirmation step will be implemented to ensure the user's intent.
- **4.1.2.4. Import Previous Data:**
  - **Summary:** This task involves allowing users to import trip data from previous years, implementing a secure file upload mechanism, validating the imported data format, and implementing a data mapping process.
  - **Description:** Users will be able to import trip data from previous years. A secure file upload mechanism will be implemented to prevent malicious uploads. The imported data format will be validated and ensured to be compatible with the app's database. A data mapping process will be implemented to match imported data fields with the app's database schema.
- **4.1.2.5. Trip Tracking:**
  - **Summary:** This task involves initiating GPS tracking, recording trip details, implementing background location tracking, requesting user permission for location access, and ensuring compliance with privacy regulations.
  - **Description:** GPS tracking will be initiated using the device's built-in GPS sensor. Trip details, including start time, start location (latitude/longitude), mode (selected by the user), and trip type (selected by the user), will be recorded. Background location tracking will be implemented to continue recording trip data even when the app is not in the foreground. User permission for location access will be requested, and compliance with privacy regulations will be ensured.
- **4.1.2.6. End Trip:**
  - **Summary:** This task involves stopping GPS tracking, calculating trip distance and duration, recording the end time and location, and updating the user's trip log.
  - **Description:** GPS tracking will be stopped. Trip distance and duration will be calculated using the recorded GPS data. The end time and end location (latitude/longitude) will be recorded. The user's trip log will be updated with the recorded trip details.
- **4.1.2.7. Trip Type Selection:**
  - **Summary:** This task involves providing a dropdown list of trip types and allowing users to select the appropriate trip type.
  - **Description:** A dropdown list of trip types (Work, Errand, Social, Other) will be provided. Users will be able to select the appropriate trip type.
- **4.1.2.8. Mode Selection:**
  - **Summary:** This task involves providing a dropdown list of transportation modes and allowing users to select the appropriate mode of transportation.
  - **Description:** A dropdown list of transportation modes (Walk, Bike, Run, E-bike, Skate, Other) will be provided. Users will be able to select the appropriate mode of transportation.
- **4.1.2.9. Manual Trip Entry:**

  - **Summary:** This task involves allowing users to manually enter trip details, implementing input validation, and using secure input methods to prevent XSS attacks.
  - **Description:** Users will be able to manually enter trip details, including date, miles, trip type, mode, start time, end time, and start and end locations (address or latitude/longitude). Input validation will be implemented to ensure data integrity and format compliance. Secure input methods will be used to prevent XSS attacks.
- **4.1.2.10. Validation:**
  - **Summary:** This task involves validating the entered data to ensure it is within acceptable ranges and formats and implementing error handling.
  - **Description:** The entered data will be validated to ensure it is within acceptable ranges and formats. Error handling will be implemented to display appropriate messages to the user in case of invalid input.
- **4.1.2.11. Trip Log Update:**
  - **Summary:** This task involves adding the manually entered trip to the user's trip log and updating the user's cumulative data.
  - **Description:** The manually entered trip will be added to the user's trip log. The user's cumulative data will be updated based on the new trip.

#### 4.1.3. Data & Analytics

- **4.1.3.1. Personal Dashboard:**
  - **Summary:** This task involves displaying cumulative data for the current season, implementing data visualization techniques, and allowing users to view data for the current calendar year and lifetime of their account.
  - **Description:** Cumulative data for the current season (Memorial Day - Columbus Day) will be displayed, including the number of trips, total mileage, gas saved (calculated based on user-defined fuel efficiency), CO2e reduced (calculated based on user-defined fuel efficiency and CO2e emissions per gallon), and money saved (calculated based on user-defined fuel price and mileage). Data visualization techniques (e.g., charts, graphs) will be implemented to present the data in an easily understandable format.
- **4.1.3.2. Calendar Year Data:**
  - **Summary:** This task involves allowing users to view cumulative data for the current calendar year and implementing data filtering and sorting options.
  - **Description:** Users will be able to view cumulative data for the current calendar year. Data filtering and sorting options will be implemented to allow users to explore the data.
- **4.1.3.3. Lifetime Data:**
  - **Summary:** This task involves allowing users to view cumulative data for the lifetime of their account and implementing data aggregation and visualization techniques.
  - **Description:** Users will be able to view cumulative data for the lifetime of their account. Data aggregation and visualization techniques will be implemented to present the data in a meaningful way.
- **4.1.3.4. Leaderboards:**
  - **Summary:** This task involves displaying a list of individual participants ranked by their cumulative data for the current season, implementing pagination and search functionality, and allowing users to filter the leaderboard.
  - **Description:** A list of individual participants ranked by their cumulative data for the current season will be displayed. The following information will be displayed for each participant: display name (username), team (if applicable), number of trips logged, total miles logged, gas saved, CO2e reduced, and money saved. Pagination and search functionality will be implemented for large leaderboards. Users will be able to filter the leaderboard by team or by specific data points.
- **4.1.3.5. Team Leaderboard:**
  - **Summary:** This task involves displaying a list of teams ranked by their cumulative data for the current season, implementing pagination and search functionality, and allowing users to view individual participants within each team.
  - **Description:** A list of teams ranked by their cumulative data for the current season will be displayed. The following information will be displayed for each team: team name, number of current members, miles logged, trips logged, CO2e reduced, and gas saved. Pagination and search functionality will be implemented for large leaderboards. Users will be able to view the individual participants within each team.
- **4.1.3.6. Historical Leaderboards:**
  - **Summary:** This task involves allowing users to view historical leaderboards from previous years and implementing data filtering and sorting options.
  - **Description:** Users will be able to view historical leaderboards from previous years. Data filtering and sorting options will be implemented to allow users to explore historical data.
- **4.1.3.7. Season Dashboard:**
  - **Summary:** This task involves displaying real-time data for the current challenge, implementing data visualization techniques, and updating the dashboard data in real-time.
  - **Description:** Real-time data for the current challenge will be displayed, including cumulative miles logged, CO2e/GHG reductions, total number of trips logged, and number of active participants. Data visualization techniques (e.g., charts, graphs) will be implemented to present the data in an easily understandable format. The dashboard data will be updated in real-time as new trips are logged.
- **4.1.3.8. Sync with Website:**
  - **Summary:** This task involves syncing the dashboard data with the dashboard on www.solepower.org and implementing a data synchronization mechanism.
  - **Description:** The dashboard data will be synced with the dashboard on www.solepower.org. A data synchronization mechanism will be implemented to ensure data consistency between the app and the website.

- **4.1.3.9. Lifetime Dashboard:**
  - **Summary:** This task involves displaying cumulative data for the lifetime of the Sole Power program and implementing data visualization techniques.
  - **Description:** Cumulative data for the lifetime of the Sole Power program will be displayed, including total miles logged, CO2e/GHG reductions, total number of trips logged, and total number of participants. Data visualization techniques (e.g., charts, graphs) will be implemented to present the data in a meaningful way.

### 4.1.4. Navigation & Information

- **4.1.4.1. Mapping & Geolocation:**
  - **Summary:** This task involves providing users with route recommendations based on their starting point and destination, integrating with a mapping API, and allowing users to customize route preferences.
  - **Description:** Users will be provided with route recommendations based on their starting point and destination. A mapping API (e.g., Google Maps, Mapbox) will be integrated to provide accurate and up-to-date route information. Users will be able to customize route preferences (e.g., shortest distance, fastest time, avoid tolls).
- **4.1.4.2. Navigation Assistance:**
  - **Summary:** This task involves offering turn-by-turn navigation guidance, integrating with a navigation API, and implementing voice guidance.
  - **Description:** Turn-by-turn navigation guidance will be offered for selected routes. A navigation API (e.g., Google Maps Navigation, Mapbox Navigation) will be integrated to provide real-time navigation instructions. Voice guidance will be implemented for hands-free navigation.
- **4.1.4.3. Bike Lanes & Walking Paths:**
  - **Summary:** This task involves displaying information about available bike lanes and walking paths, integrating with a mapping API, and allowing users to filter the map.
  - **Description:** Information about available bike lanes and walking paths will be displayed. A mapping API will be integrated to retrieve and display bike lane and walking path data. Users will be able to filter the map by bike lanes, walking paths, or both.
- **4.1.4.4. Public Transit Routes:**
  - **Summary:** This task involves providing information about public transit routes and schedules, integrating with a public transit API, and allowing users to search for specific routes or stops.
  - **Description:** Information about public transit routes and schedules will be provided. A public transit API (e.g., Transitland, Moovit) will be integrated to retrieve and display real-time transit information. Users will be able to search for specific routes or stops.
- **4.1.4.5. Program Information:**
  - **Summary:** This task involves displaying information about the Sole Power program, its goals, and its history, including program guidelines, rules, and regulations, and providing contact information.
  - **Description:** Information about the Sole Power program, its goals, and its history will be displayed. Program guidelines, rules, and regulations will be included. Contact information for the Sole Power program will be provided.
- **4.1.4.6. Incentives & Rewards:**
  - **Summary:** This task involves providing information about available incentives and rewards for participants, including details about eligibility criteria, reward levels, and redemption processes.
  - **Description:** Information about available incentives and rewards for participants will be provided. Details about eligibility criteria, reward levels, and redemption processes will be included.
- **4.1.4.7. News & Updates:**
  - **Summary:** This task involves displaying news and updates about the Sole Power program and implementing a content management system.
  - **Description:** News and updates about the Sole Power program will be displayed. A content management system will be implemented to allow administrators to easily update news and announcements.
- **4.1.4.8. Contact Information:**
  - **Summary:** This task involves displaying contact information for the Sole Power program and including a link to the program's website.
  - **Description:** Contact information for the Sole Power program will be displayed. A link to the program's website will be included.
- **4.1.4.9. Contact Form:**
  - **Summary:** This task involves allowing users to submit inquiries or feedback, implementing a secure contact form, and integrating the contact form with the app's backend.
  - **Description:** Users will be able to submit inquiries or feedback. A secure contact form will be implemented to prevent spam and malicious submissions. The contact form will be integrated with the app's backend to send notifications to administrators.

### 4.1.5. Social Features

- **4.1.5.1. Social Sharing:**
  - **Summary:** This task involves allowing users to share their progress and achievements on social media platforms, implementing social media integration, and allowing users to customize the sharing message and content.
  - **Description:** Users will be able to share their progress and achievements on social media platforms (e.g., Facebook, Twitter, Instagram). Social media integration will be implemented using APIs to facilitate sharing. Users will be able to customize the sharing message and content.
- **4.1.5.2. Peer-to-Peer Encouragement:**
  - **Summary:** This task involves enabling users to send messages of support to other participants, implementing a messaging system,

    and allowing users to send private or public messages.
  - **Description:** Users will be able to send messages of support to other participants. A messaging system within the app will be implemented to facilitate communication between users. Users will be able to send private messages or public messages to specific teams or groups.
- **4.1.5.3. Team Chat:**
  - **Summary:** This task involves providing a platform for team members to communicate with each other, implementing a team chat feature, and allowing users to send messages, share images, and create group chats.
  - **Description:** A platform for team members to communicate with each other will be provided. A team chat feature will be implemented within the app. Users will be able to send messages, share images, and create group chats.

**4.1.6. Push Notifications**

- **4.1.6.1. Program Updates:**
  - **Summary:** This task involves sending notifications about new program features, events, or announcements, implementing a push notification system, and allowing users to customize their notification settings.
  - **Description:** Notifications about new program features, events, or announcements will be sent. A push notification system will be implemented using a third-party service (e.g., Firebase Cloud Messaging, OneSignal). Users will be able to customize their notification settings.
- **4.1.6.2. Challenge Reminders:**
  - **Summary:** This task involves reminding participants about upcoming challenges and deadlines and implementing scheduled push notifications.
  - **Description:** Participants will be reminded about upcoming challenges and deadlines. Scheduled push notifications will be implemented to remind users about important dates and events.
- **4.1.6.3. Personal Progress Updates:**
  - **Summary:** This task involves notifying users about their progress towards their goals and implementing personalized push notifications.
  - **Description:** Users will be notified about their progress towards their goals. Personalized push notifications will be implemented to keep users engaged and motivated.

**4.1.7. Accessibility**

- **4.1.7.1. Language Support:**
  - **Summary:** This task involves providing the app in English and Spanish and implementing language localization.
  - **Description:** The app will be provided in English and Spanish. Language localization will be implemented to ensure that the app is accessible to users who speak different languages.
- **4.1.7.2. User Interface Design:**
  - **Summary:** This task involves ensuring the app is user-friendly and accessible to all users, following accessibility guidelines, implementing a clear and consistent user interface design, using high-contrast colors and fonts, and providing alternative input methods.
  - **Description:** The app will be user-friendly and accessible to all users, regardless of their age or technical ability. Accessibility guidelines (e.g., WCAG) will be followed to ensure that the app is usable by people with disabilities. A clear and consistent user interface design will be implemented. High-contrast colors and fonts will be used to improve readability. Alternative input methods (e.g., voice control, gesture recognition) will be provided for users with disabilities.

**4.2. Administrator Role**

**4.2.1. User Management**

- **4.2.1.1. User List:**
  - **Summary:** This task involves displaying a list of all registered users, implementing pagination and search functionality, and allowing administrators to filter users.
  - **Description:** A list of all registered users will be displayed, including their profile information: user ID, first name, last name, email address, team (if applicable), date of birth (if provided), phone number (if provided), account status (Active, Deactivated), date created, and last login date. Pagination and search functionality will be implemented for large user lists. Administrators will be able to filter users by team, account status, or other criteria.
- **4.2.1.2. User Search:**
  - **Summary:** This task involves allowing administrators to search for specific users by user ID, first name, last name, email address, or team name and implementing a robust search algorithm.
  - **Description:** Administrators will be able to search for specific users by user ID, first name, last name, email address, or team name. A robust search algorithm will be implemented to ensure accurate and efficient results.
- **4.2.1.3. User Edit:**
  - **Summary:** This task involves allowing administrators to edit user profile information, implementing input validation, and using secure input methods to prevent XSS attacks.
  - **Description:** Administrators will be able to edit user profile information, including first name, last name, email address (subject to validation), password (subject to complexity requirements), date of birth, phone number, team (subject to team management rules), and account status (Active, Deactivated). Input validation will be implemented to ensure data integrity and format compliance. Secure

input methods will be used to prevent XSS attacks.

- **4.2.1.4. User Deactivation:**
  - **Summary:** This task involves allowing administrators to deactivate user accounts, implementing a confirmation step, and preventing deactivated users from logging in or accessing the app.
  - **Description:** Administrators will be able to deactivate user accounts. A confirmation step will be implemented to ensure the administrator's intent. User accounts will be deactivated by setting their account status to "Deactivated". Deactivated users will be prevented from logging in or accessing the app.
- **4.2.1.5. User Reactivation:**
  - **Summary:** This task involves allowing administrators to reactivate deactivated user accounts, implementing a confirmation step, and allowing reactivated users to log in and access the app.
  - **Description:** Administrators will be able to reactivate deactivated user accounts. A confirmation step will be implemented to ensure the administrator's intent. User accounts will be reactivated by setting their account status to "Active". Reactivated users will be allowed to log in and access the app.

#### 4.2.2. Team Management

- **4.2.2.1. Team List:**
  - **Summary:** This task involves displaying a list of all registered teams, implementing pagination and search functionality, and allowing administrators to filter teams.
  - **Description:** A list of all registered teams will be displayed, including team ID, team name, team description, number of members, team status (Active, Pending, Rejected), and date created. Pagination and search functionality will be implemented for large team lists. Administrators will be able to filter teams by status or other criteria.
- **4.2.2.2. Team Creation:**
  - **Summary:** This task involves allowing administrators to create new teams, implementing input validation, and setting the team status to "Active" upon creation.
  - **Description:** Administrators will be able to create new teams by providing a unique team name and a team description. Input validation will be implemented to ensure data integrity and format compliance. The team status will be set to "Active" upon creation.
- **4.2.2.3. Team Approval:**
  - **Summary:** This task involves allowing administrators to approve or reject team creation requests, implementing a confirmation step, and updating the team status accordingly.
  - **Description:** Administrators will be able to approve or reject team creation requests. A confirmation step will be implemented to ensure the administrator's intent. Approved teams will have their status set to "Active". Rejected teams will have their status set to "Rejected".
- **4.2.2.4. Team Edit:**
  - **Summary:** This task involves allowing administrators to edit team details, implementing input validation, and using secure input methods to prevent XSS attacks.
  - **Description:** Administrators will be able to edit team details, including team name, team description, and team status (Active, Pending, Rejected). Input validation will be implemented to ensure data integrity and format compliance. Secure input methods will be used to prevent XSS attacks.
- **4.2.2.5. Team Deletion:**
  - **Summary:** This task involves allowing administrators to delete teams, implementing a confirmation step, and removing all associated users from the team.
  - **Description:** Administrators will be able to delete teams. A confirmation step will be implemented to ensure the administrator's intent. All associated users will be removed from the team upon deletion.

#### 4.2.3. Trip Management

- **4.2.3.1. Trip List:**
  - **Summary:** This task involves displaying a list of all logged trips, implementing pagination and search functionality, and allowing administrators to filter trips.
  - **Description:** A list of all logged trips will be displayed, including trip ID, date, miles, trip type, mode, start time, end time, start location (latitude/longitude), end location (latitude/longitude), and user ID. Pagination and search functionality will be implemented for large trip lists. Administrators will be able to filter trips by date, trip type, mode, or user ID.
- **4.2.3.2. Trip Search:**
  - **Summary:** This task involves allowing administrators to search for specific trips by trip ID, date, miles, trip type, mode, start location, end location, or user ID and implementing a robust search algorithm.
  - **Description:** Administrators will be able to search for specific trips by trip ID, date, miles, trip type, mode, start location, end location, or user ID. A robust search algorithm will be implemented to ensure accurate and efficient results.
- **4.2.3.3. Trip Edit:**
  - **Summary:** This task involves allowing administrators to edit trip details, implementing input validation, and using secure input methods to prevent XSS attacks.
  - **Description:** Administrators will be able to edit trip details, including date, miles, trip type, mode, start time, end time, start location, and end location. Input validation will be implemented to ensure data integrity and format compliance. Secure input methods will be used to prevent XSS attacks.
- **4.2.3.4. Trip Deletion:**
  - **Summary:** This task involves allowing administrators to delete trips, implementing a confirmation step, and removing the trip from the

user's trip log.
  - **Description:** Administrators will be able to delete trips. A confirmation step will be implemented to ensure the administrator's intent. The trip will be removed from the user's trip log upon deletion.

#### 4.2.4. Data & Analytics

- **4.2.4.1. Dashboard Access:**
  - **Summary:**